

Figure 1

201	76 Box Rd Townsville QLD
202	PO Box 92 Geelong VIC
203	39 Main St Box Hill VIC
204	8 Box Ave Devonport TAS
205	Cnr Box and Wolger Rds Townsville QLD
206	76 Box St Townsville NSW
207	231 Box Road Townsville QLD
208	53 3rd Ave, Townsville 4321 QLD
209	35 Third Avenue, Townsville Queensland 4321
210	333 Mt Pleasant Road, Springvale
211	191 Springvale Road, Mt Pleasant
212	123 Sydney Ave, Melbourne VIC

Figure 2

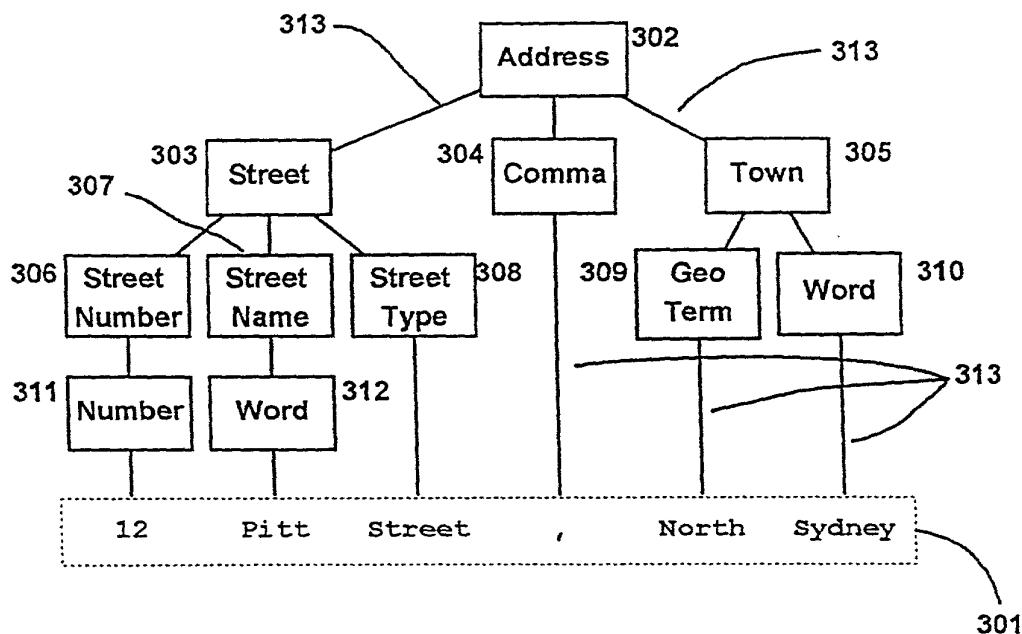


Figure 3

Address	<input type="text"/>
Address 1	<input type="text"/>
Address 2	<input type="text"/>
Address 3	<input type="text"/>
Address 4	<input type="text"/>
Address 5	<input type="text"/>
Address 1	<input type="text"/>
Address 2	<input type="text"/>
Suburb	<input type="text"/>
State	<input type="text"/>
Postcode	<input type="text"/>
Unit #	<input type="text"/>
Street Name	<input type="text"/>
Town	<input type="text"/>
State	<input type="text"/>
Postcode	<input type="text"/>

Figure 4

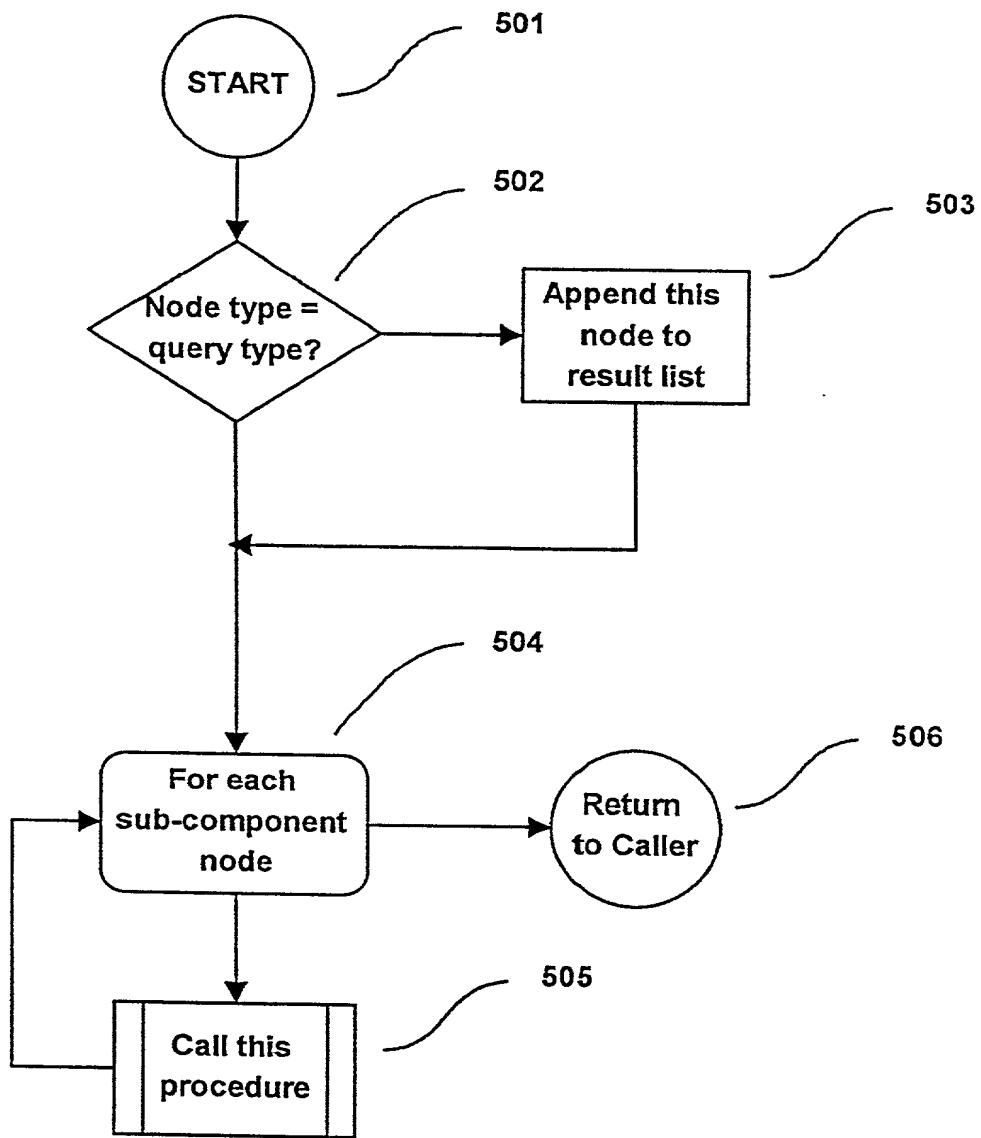


Figure 5

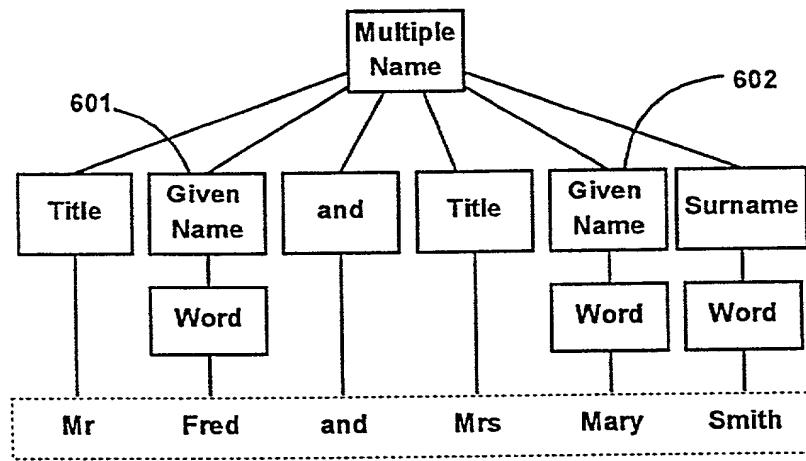


Figure 6

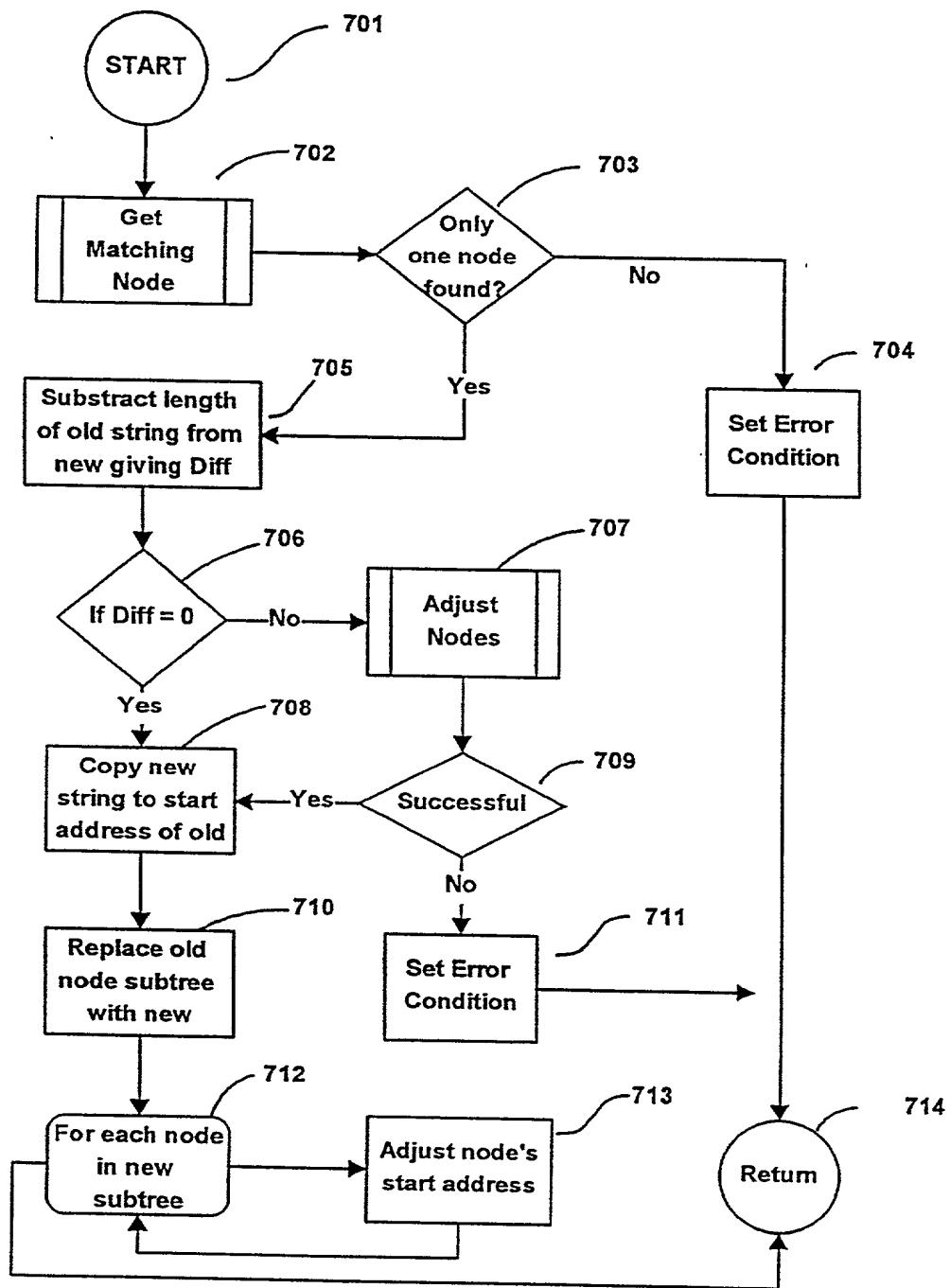


Figure 7

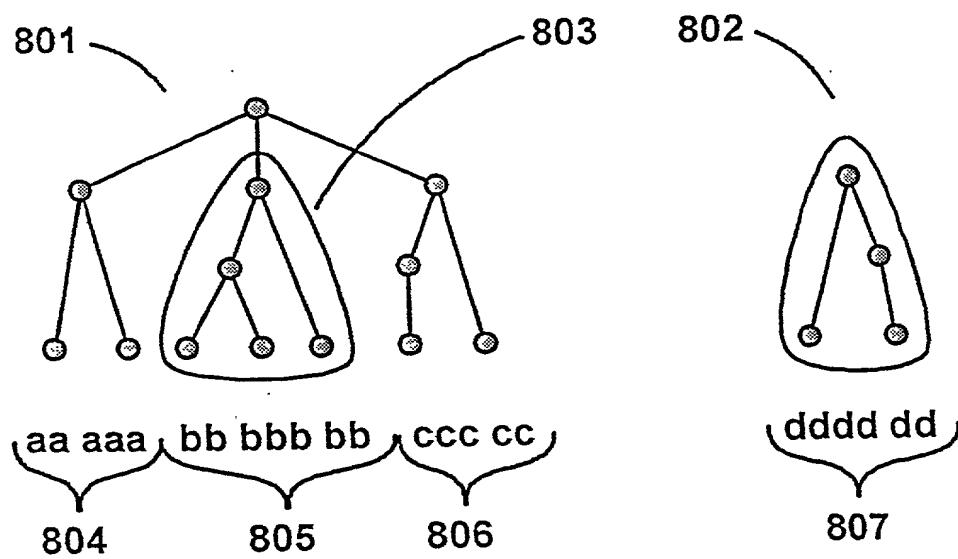


Figure 8

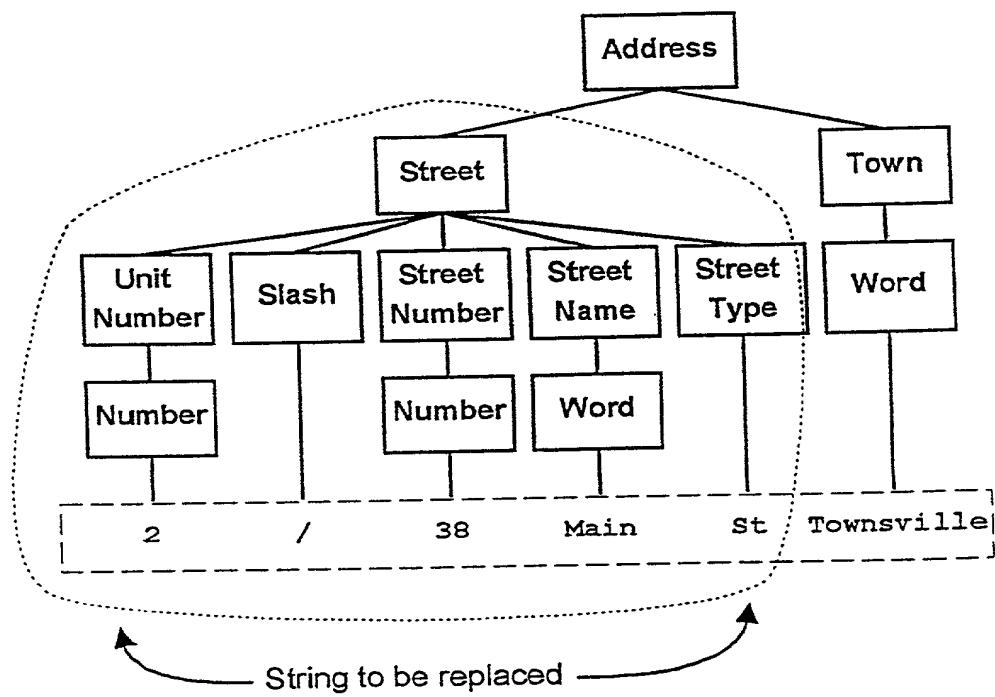


Figure 9

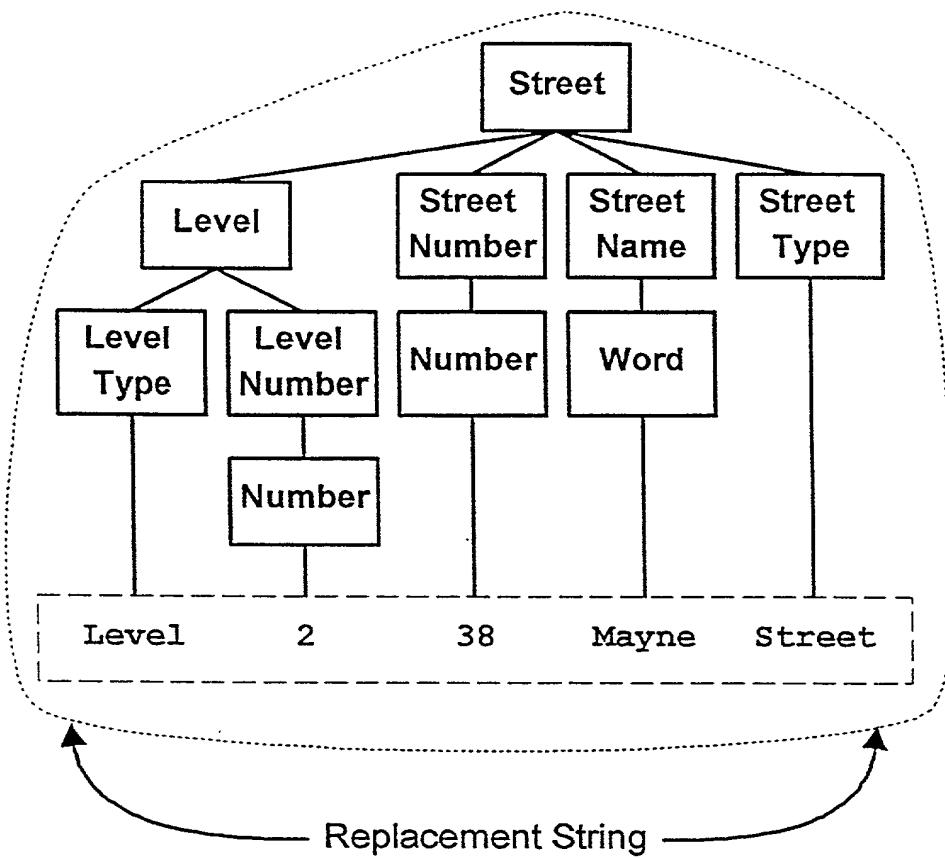


Figure 10

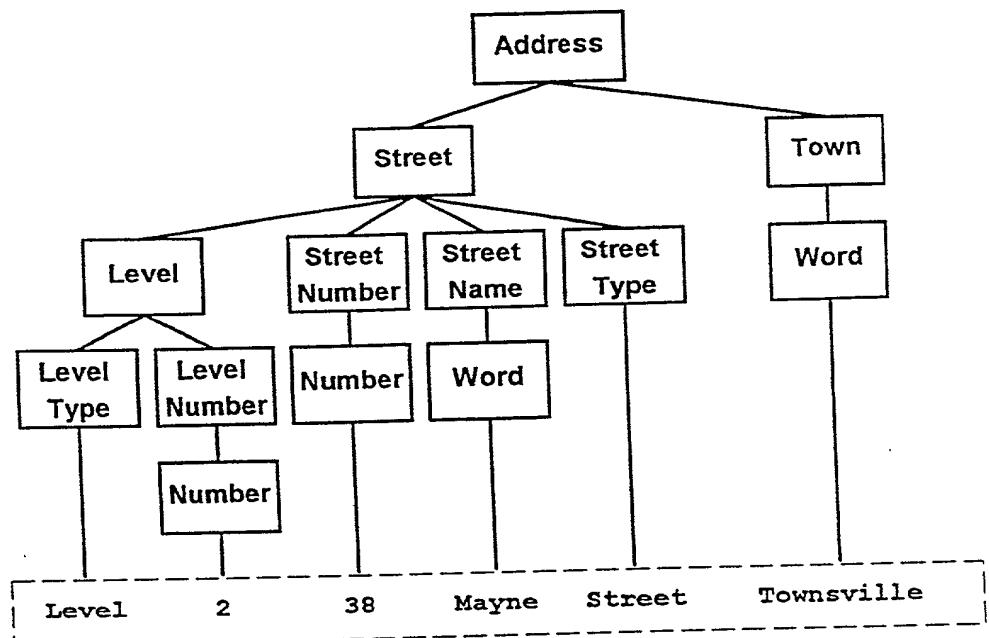


Figure 11

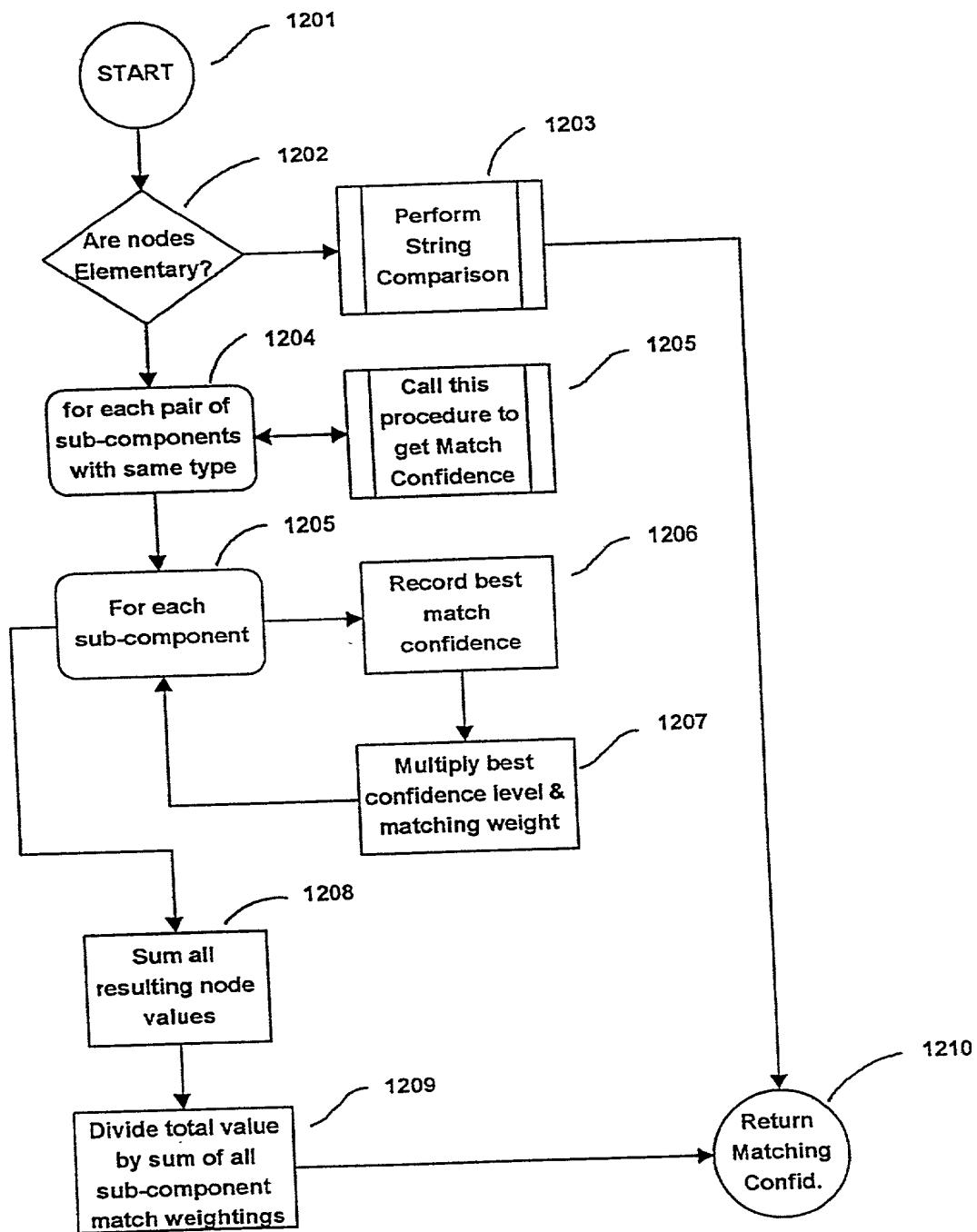


Figure 12

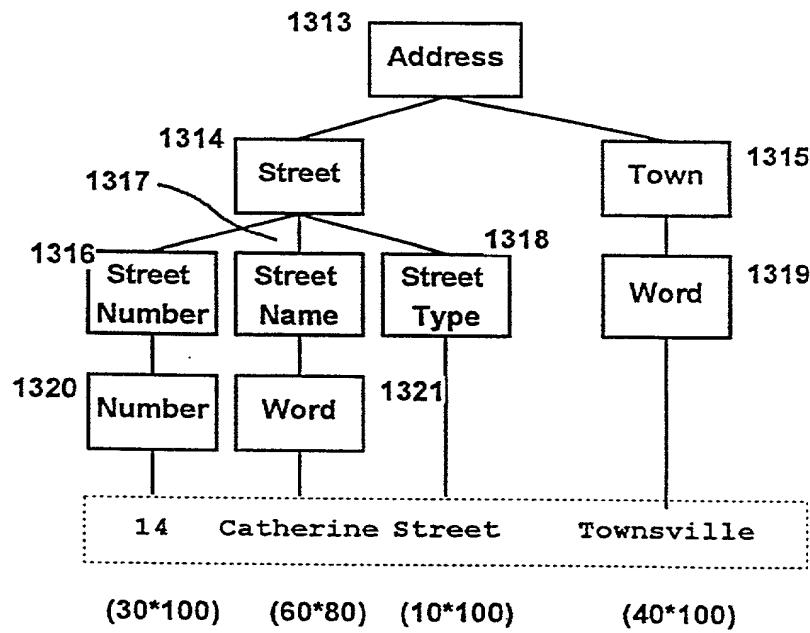
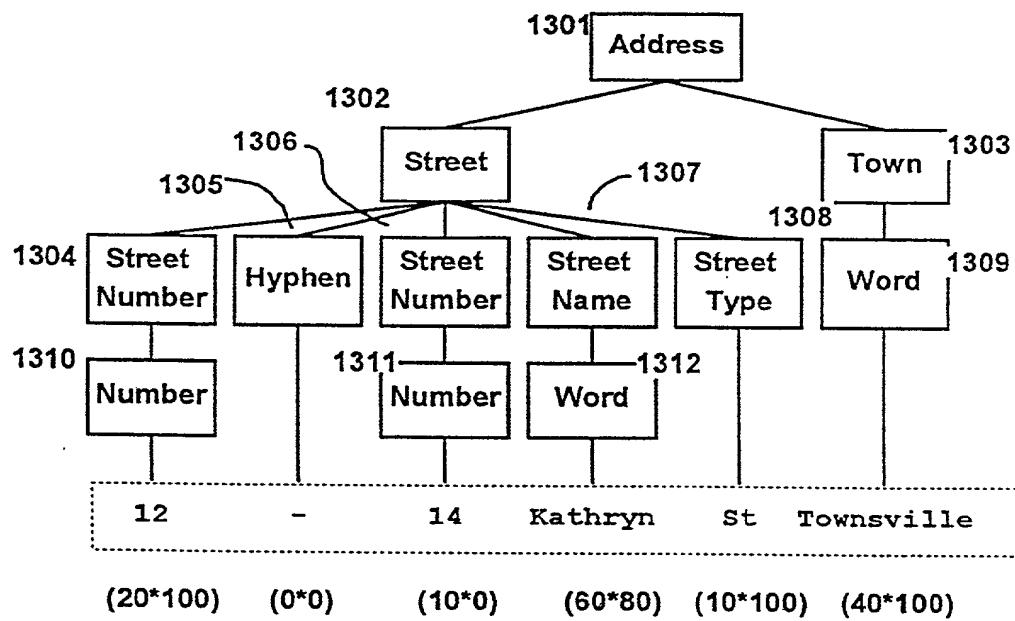


Figure 13

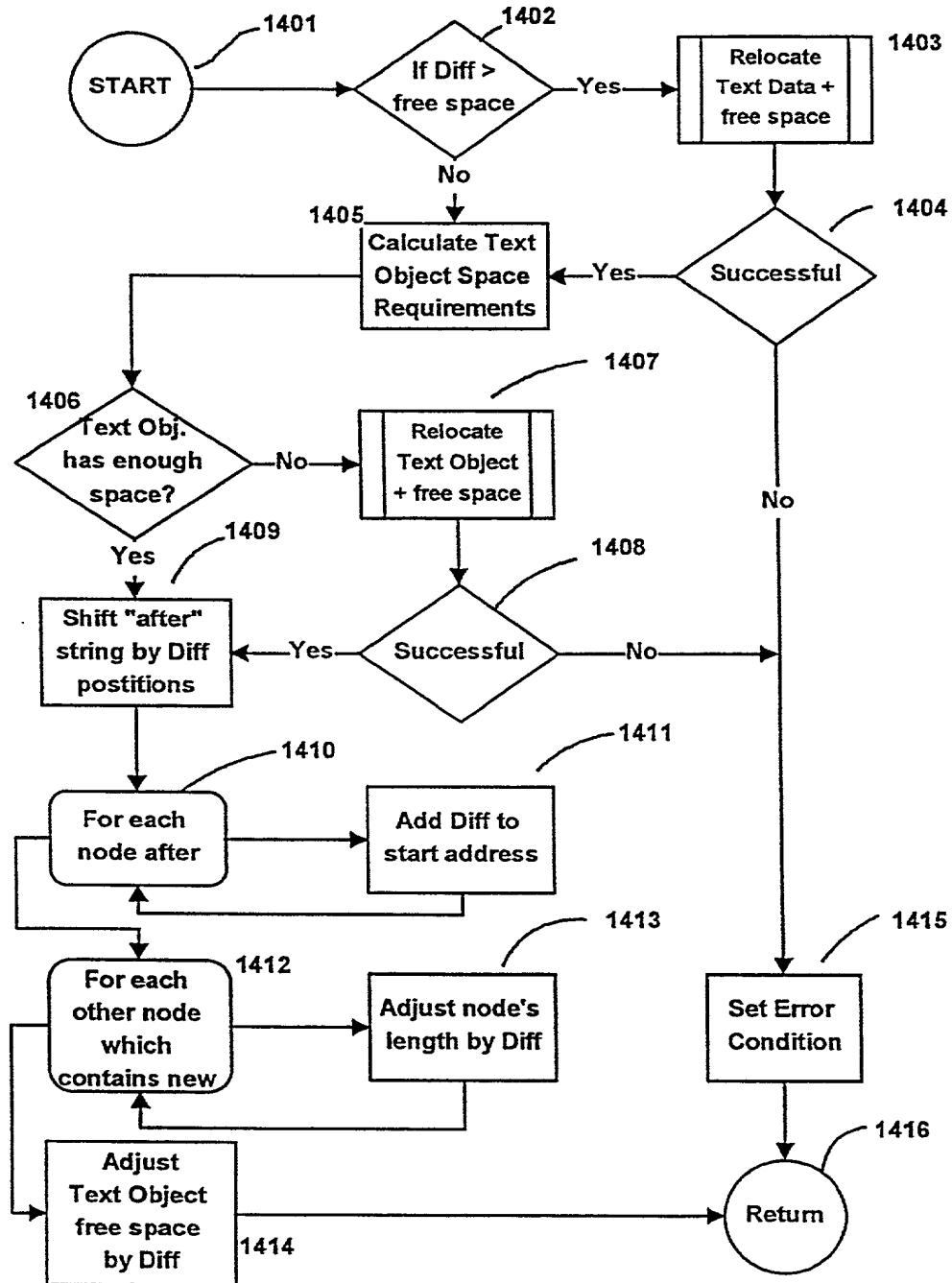


Figure 14

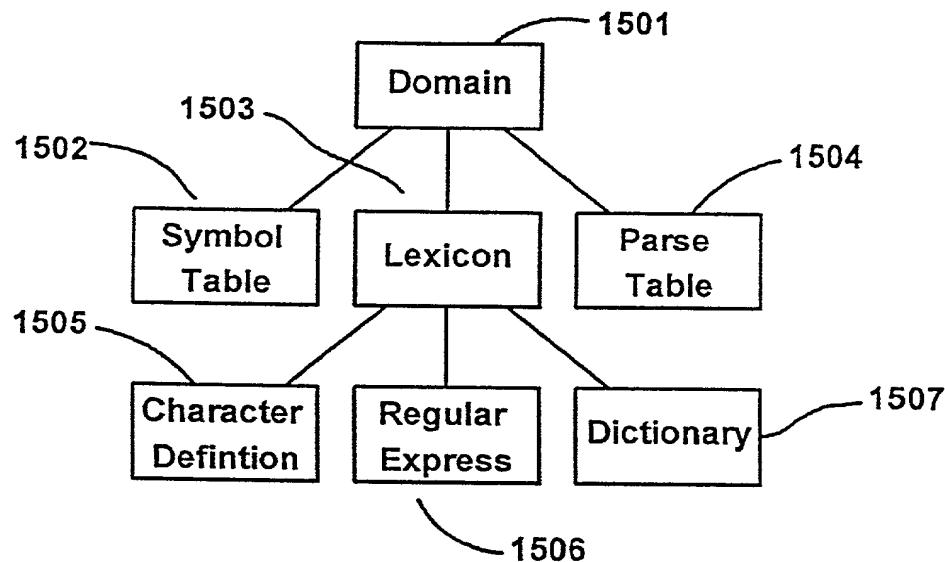


Figure 15

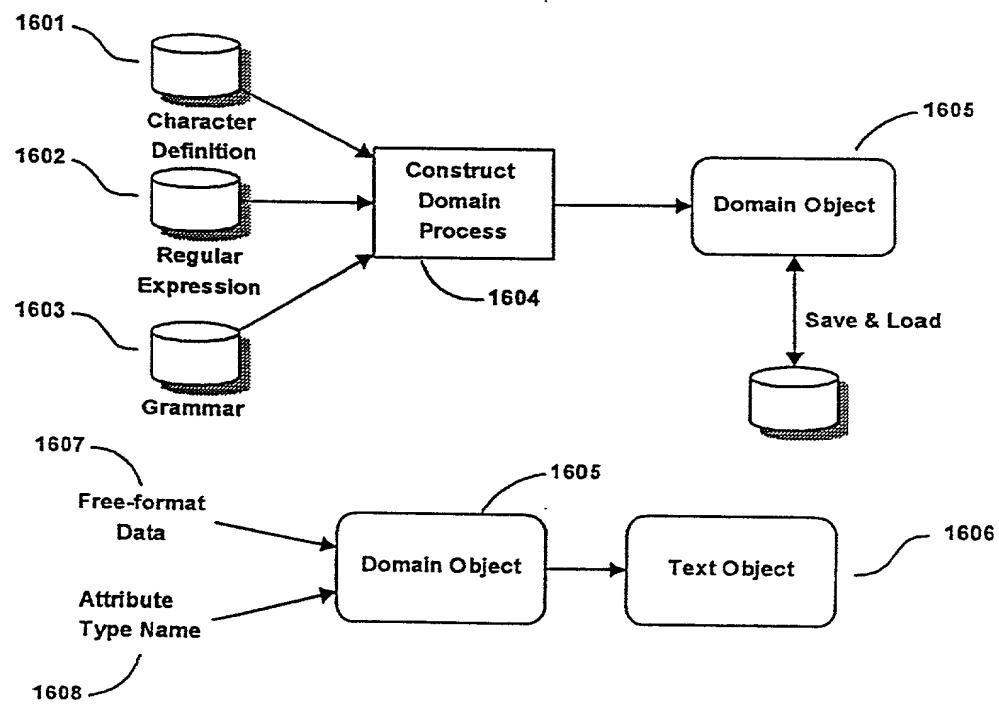


Figure 16

## Standard Japanese Katakana Transliteration

a ア	i イ	u ウ	e エ	o オ
ka カ	ki キ	ku ク	ke ケ	ko コ
sa サ	si シ	su ス	se セ	so ソ
ta タ	ti チ	tu ツ	te テ	to ト
na ナ	ni ニ	nu ヌ	ne ネ	no ノ
ha ハ	hi ヒ	hu フ	he ヘ	ho ホ
ma マ	mi ミ	mu ム	me メ	mo モ
ya ヤ		yu ュ		yo ヨ
ra ラ	ri リ	ru ル	re レ	ro ロ
wa ワ	wi ヲ		we ウエ	wo ヲ
ga ガ	gi ギ	gu グ	ge ゲ	go ゴ
za ザ	zi ジ	zu ズ	ze ゼ	zo ゾ
da ダ	di ヂ	du ド	de デ	do ド
ba バ	bi ビ	bu ブ	be ベ	bo ボ
pa パ	pi ピ	pu プ	pe ペ	po ポ
n ナ				

## Standard Greek Transliteration

A α α	I ι ι	P ρ ρ
B β β	K κ κ	S σ σ
Γ γ γ	Λ λ λ	T τ τ
Δ δ δ	Μ μ μ	Υ υ υ
Ε ε ε	Ν ν ν	Φ φ φ
Ζ ζ ζ	Ξ ξ ξ	Χ χ χ
Η η η	Ο ο ο	Ψ ψ ψ
Θ θ θ	Π π π	Ω ω ω

Figure 17

## Sample Regular Expression Definition .

		Action **	alpha	digit	symbol	space	end of line	end of string
0	Error	0						
1	start	1	3	9	13	1	16	18
2	empty	4						
3	Initial	2	6	5	5	4	5	5
4	Initial space+	3	5	5	5	4	5	5
5	Initial ^space	5						
6	Word+	2	6	8	8	7	8	8
7	Word+ space	3	8	8	8	7	8	8
8	Word+ ^space	7						
9	0-9	2	12	10	12	11	12	12
10	0-9+	2	12	10	12	11	12	12
11	0-9+ space	3	12	12	12	11	12	12
12	0-9+ ^space	8						
13	sym	2	15	15	15	14	15	15
14	sym space	3	15	15	15	14	15	15
15	sym ^space	10						
16	eol+ space*	3	17	17	17	16	16	18
17	eol+ ^space	11						
18	eoi	9	0	0	0	0	0	0

** Action	Action Description
0	Error in Table
1	Bypass leading spaces
2	Append this character to character buffer
3	Append trailing space to character buffer
4	Empty string
5	Create "initial" token; go back 1 char; set state to 1
7	Create "word" token; go back 1 char; set state to 1
8	Create "number" token; go back 1 char; set state to 1
9	Create "end of input" token; go back 1 char; set state to 1
10	Create "symbol" token; go back 1 char; set state to 1
11	Create "end of line" token; go back 1 char; set state to 1

Figure 18

```

Address
-> StreetAddr, Town Zipcode State
| PostBox, Town Zipcode State ;
StreetAddr
-> Street
| StreetNum Street
| AptType AptNum StreetNum Street
| StreetNum Street AptType AptNum ;
Street
-> StreetName StreetType StreetDir      :-2
| StreetName StreetType ;
StreetName
-> word | word word ;
StreetNum -> nbr ;
AptNum -> nbr ;
StreetType
-> "Ave"  | "Avenue" ("Ave")
| "Rd"    | "Road"    ("Rd")
| "St"    | "Street"  ("St") ;
StreetDir
-> Geo ;
Geo
-> "North" | "N" ("North")
| "South"  | "S" ("South")
| "East"   | "E" ("East")
| "West"   | "W" ("West") ;
AptType
-> "Apt"  | "Apartment"
| "Unit"
| "Suite" | "Ste" ;
Zipcode
-> 99999  | 99999 "-" 9999 ;
PostBox
-> PostPref PostBoxNum ;
PostPref
-> "PO Box" | "Box" ;      ==9=
PostBoxNum
-> nbr | nbr A | A nbr ;
Town
-> word | word word
| Geo word | Geo word word ;
State
-> "ALABAMA"    ("AL") | "AL"
| "ALASKA"      ("AK") | "AK"
| "ARIZONA"     ("AZ") | "AZ"
| "ARKANSAS"    ("AR") | "AR"
| "CALIFORNIA"  ("CA") | "CA" ;

```

Special Symbols:

->	Consists of
	Or
:	Rule terminator
(..)	Matching equivalence
=+9=	Matching Significance Adjustment
==	Zero Matching Significance
:+9	Parsing Significance Adjustment

Reserved Words:

word	one or more letters
nbr	one or more digits
A	one letter
9	one digit
,	comma or nothing

Figure 19

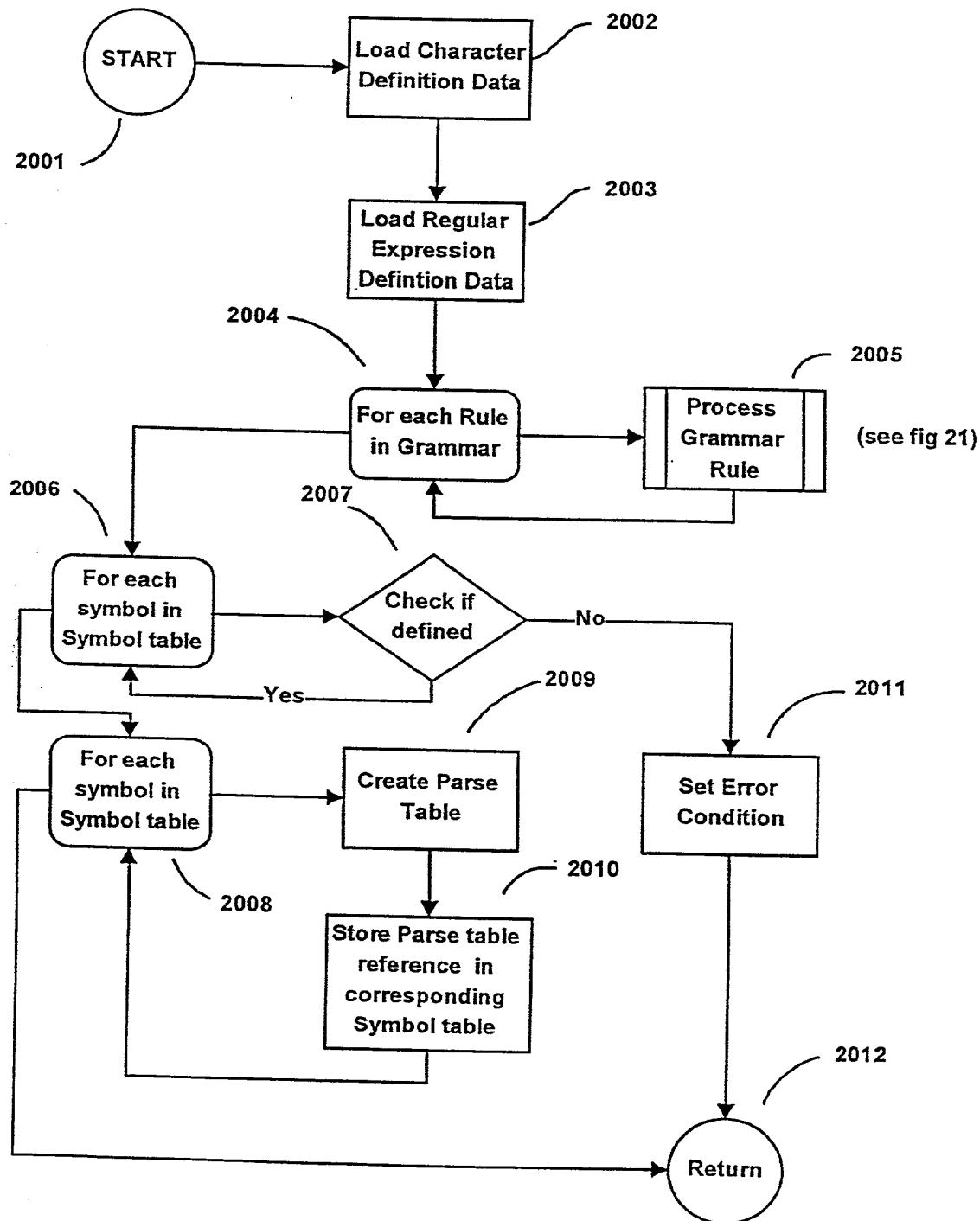


Figure 20

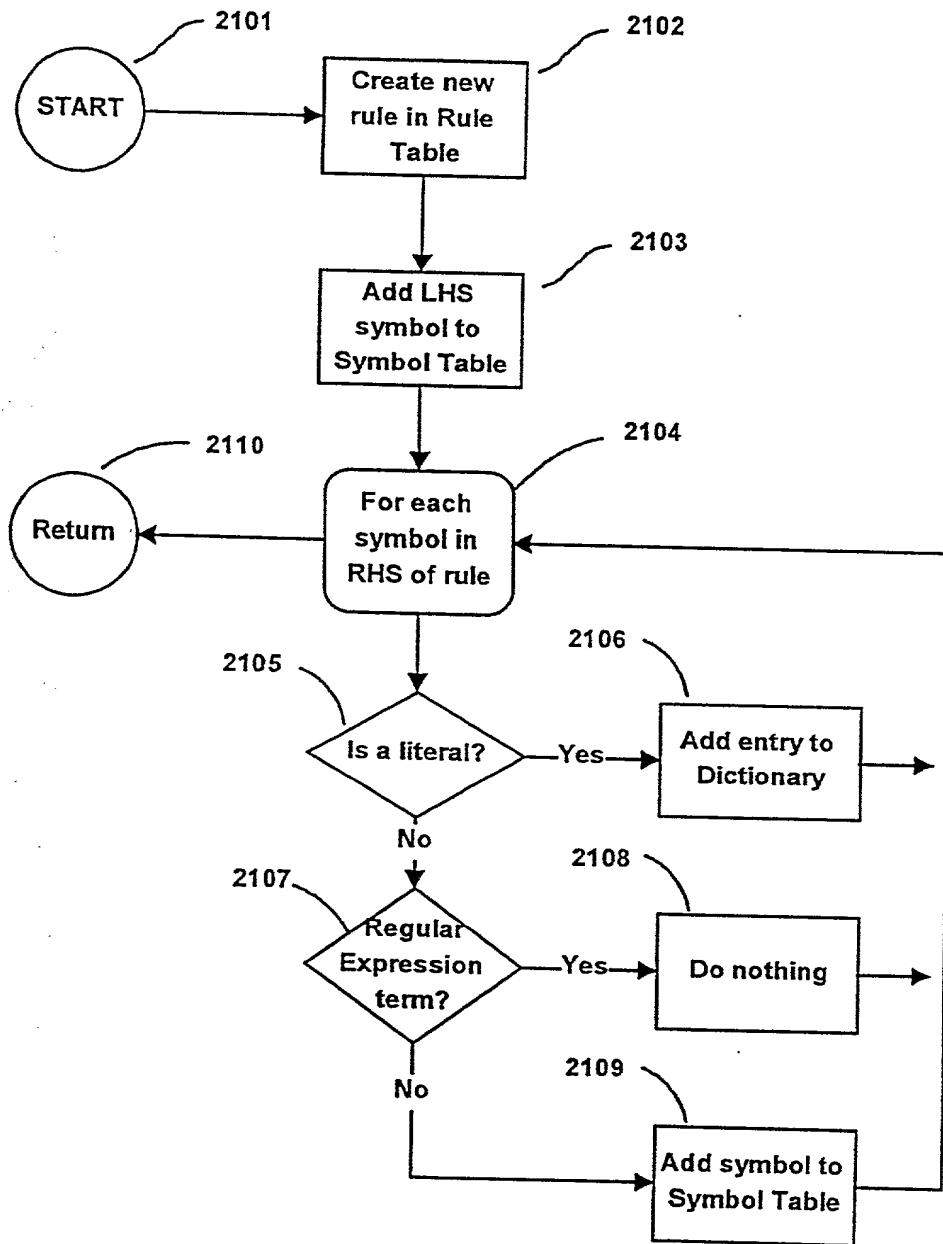


Figure 21

SQL Database Implementation Example

```
1. CREATE DOMAINOBJECT US_ADDRESS ;
2. UPDATE US_ADDRESS
   SET LANGUAGE = EXTERNAL 'Path/English.txt',
       GRAMMAR = EXTERNAL 'Path/US_Addr.txt' ;

3. CREATE TEXTOBJECT ADDRESS ;
4. UPDATE US_ADDRESS
   SET DOMAIN = US_ADDRESS,
       TYPE = "Address" ;

5. CREATE TABLE PERSONS(
   Name          CHAR (20),
   Home_addr     ADDRESS ) ;

6. INSERT INTO PERSONS ( Name, Home_addr )
   VALUES (
   "John Smith",
   "123 Cathy Street, Apt 5, Huntsvale, CA, 98765" ) ;

7. SELECT FROM PERSONS
   WHERE Home_addr = "Unit 5 123 Cathy St, Huntsvale, CA" ;

8. SELECT FROM PERSONS
   WHERE Home_addr.State = "California" ;

9. SELECT FROM PERSONS
   WHERE Home_addr.Street MATCHES "Kathie St" > 0.80 ;
```

New Reserved Words:

DOMAINOBJECT, TEXTOBJECT, LANGUAGE,  
GRAMMAR, TYPE, MATCHES

Figure 22